

Resum

El present treball consisteix en el desenvolupament d'una eina per el seguiment de marcadors al llarg d'un vídeo que captura algun moviment del cos humà. Els marcadors s'utilitzen en biomecànica per identificar punts d'interès en un moviment. L'objectiu del software desenvolupat en el projecte és facilitar l'estudi d'aquests punts.

El funcionament del programa desenvolupat s'emmarca en l'entorn MATLAB. A partir d'un vídeo on es mostra el moviment a estudiar s'identifiquen els marcadors i se'n fa el seguiment. Aquest consisteix en l'obtenció de la nova posició del punt mitjançant una recerca exhaustiva en una regió centrada a la posició inicial i usant el mètode de la correlació creuada normalitzada per a la comparació dels possibles candidats a marcador en cadascun dels fotogrames que conformen la gravació.

Com a resultat del projecte s'ha obtingut un programa que proporciona el recorregut i la variació de l'angle entre els marcadors determinats per l'investigador del moviment, per al posterior anàlisi d'aquests paràmetres per part del mateix usuari.

Basant-nos en els resultats obtinguts en aquest treball es pot concloure que els mètodes utilitzats en el programa són adequats per al seguiment i reconeixement d'un patró determinat en una imatge.

Sumari

RESUM	1
SUMARI	3
1. PREFACI	5
1.1. Origen del projecte	5
1.2. Motivació	6
1.3. Requeriments previs	6
2. INTRODUCCIÓ	7
2.1. Objectius del projecte.....	7
2.2. Abast del projecte	8
3. L'ÚS DE MARCADORS EN BIOMECÀNICA	9
3.1. L'estudi de la cinemàtica en el treball.....	10
4. PROCESSAMENT D'IMATGES	12
5. METODOLOGIA PEL SEGUIMENT DELS MARCADORS	15
5.1. Mètode de cerca	17
5.1.1. Recerca exhaustiva	18
5.1.2. Algoritmes sub-optimals	19
5.1.3. Algoritme de cerca emprat en el seguiment de marcadors	20
5.2. Criteri de comparació	20
5.3. Consideracions sobre la variació de l'objecte a seguir	22
6. DESENVOLUPAMENT DEL SOFTWARE I METODOLOGIA	25
6.1. Entorn de desenvolupament.....	25
6.2. Fases del desenvolupament.....	25
6.3. Disseny inicial.....	25
6.3.1. Inicialització.....	26
6.3.2. Seguiment del marcador.....	27
6.4. Mètode de recerca	28
6.5. Criteri de comparació	29
6.6. Disseny final	30
6.6.1. Inicialització.....	30
6.6.2. Consideració de més d'un marcador	31
6.6.3. Extracció del primer fotograma i identificació del marcador	31

6.6.4. Seguiment del marcador	32
6.6.5. Emmagatzematge dels resultats	33
6.6.6. Tractament dels resultats	34
6.7. Validació del programa i resultats	37
6.8. Entorn de funcionament del programa.....	39
7. PRESSUPOST	41
8. IMPACTE AMBIENTAL	43
CONCLUSIONS	45
AGRAÏMENTS	47
BIBLIOGRAFIA.....	48
Referències bibliogràfiques.....	48
Bibliografia complementària.....	49

1. Prefaci

1.1. Origen del projecte

El projecte del desenvolupament de software per al seguiment de marcadors al llarg d'un vídeo s'origina en la tesi doctoral de la professora de fisioteràpia de la UManresa, facultat pertanyent a la Universitat de Vic – Universitat Central de Catalunya, Olga Borao. La tesi en qüestió [1], es va realitzar a la Universitat de Lleida i es va presentar el 2015.

La tesi es centra en els esquinços dels lligaments del turmell, una de les lesions de les extremitats inferiors més comunes i que posteriorment pot derivar en l'alteració de la correcta biomecànica de l'extremitat. Entre d'altres conseqüències, els individus que han patit aquesta lesió són subjectes a mostrar alteracions en els patrons d'activació muscular de tota la musculatura de la cama. En aquest sentit en la tesi s'estudia, entre d'altre coses, la relació entre l'activació de la musculatura estabilitzadora del turmell i la corresponent a la pròpia de la pelvis.

Amb aquest objectiu es va realitzar un estudi experimental **Error! No s'ha trobat l'origen de la referència.** [14] en el que els participants eren sotmesos a una situació d'estrès consistent en la pèrdua del suport en una extremitat. Per conèixer el nivell d'activació de la musculatura es va considerar l'activitat elèctrica que s'hi produeix, també anomenada activitat electromiogràfica (EMG). L'electromiografia consisteix en l'aplicació d'elèctrodes sobre els músculs que es volen estudiar, tal i com s'observa en la captura d'un dels vídeos desenvolupats en l'estudi (Fig 1.1).



Fig. 1.1: Mostra de la instrumentació emprada en l'estudi: Marcadors a la pelvis, elèctrode de referència al canell i elèctrodes superficials a diversos músculs

1.2. Motivació

La principal motivació en la realització d'aquest projecte ha sigut el repte de crear un programa amb la finalitat de solucionar un problema d'un àmbit no necessàriament emmarcat en l'enginyeria com és, en el cas que ens ocupa, la fisioteràpia tot. Aquest fet, juntament amb la necessitat de realitzar processament de imatges, ha provocat l'endinsament de l'estudiant en una matèria que no s'imparteix a l'escola i que per tant li era desconeguda.

Per altra banda, és indubtable la necessitat i la importància del desenvolupament de software en la societat de tecnologies en la que vivim, la qual cosa ha suposat un al·licient en l'execució del treball.

1.3. Requeriments previs

Per a la realització d'aquest projecte s'ha requerit de coneixements i competències bàsics d'informàtica i programació, que han sigut adquirits al llarg del grau i del desenvolupament personal de l'estudiant, així com nocions bàsiques de matemàtiques.

2. Introducció

2.1. Objectius del projecte

El projecte que ens ocupa té com a principal objectiu la creació d'un programa informàtic que analitzi un moviment determinat del cos humà enregistrat en un vídeo. En el marc de l'elaboració del programa s'estableixen una sèrie d'objectius secundaris a partir dels quals és possible assolir aquesta finalitat. Aquests s'enumeren a continuació:

- Processar un vídeo per poder treballar amb les imatges de cada fotograma.
- Permetre la interacció amb l'usuari per conèixer la posició del marcador.
- Desenvolupar un algorisme de seguiment d'un patró per tal d'aplicar-lo al marcador establert per l'usuari.
- Extreure i recopilar les dades de posició del marcador.
- Realitzar un tractament superficial de les dades obtingudes.

Tenint en compte que aquesta és la finalitat del treball, a continuació es descriuen els objectius específics o funcionals del projecte, és a dir, els diferents requisits i funcionalitats que s'han de tenir en compte en el programa entès com una eina que ha de ser utilitzada per un usuari.

- El programa ha de tenir un temps d'execució mínim i assequible.
- La interacció amb l'usuari ha de facilitar la feina de l'operador, expressant-se amb claredat i donant instruccions concises, per tal que aquest pugui fer ús del programa sense tenir coneixements previs de programació ni estar familiaritzat amb l'entorn MATLAB.
- El programa ha de proporcionar els resultats amb una visualització prèvia explicativa.

2.2. Abast del projecte

Aquest treball es centra en l'estudi de la posició de marcadors el moviment dels quals estigui comprès en un pla, i sigui capturat per vídeo de manera que aquest pla sigui paral·lel a l'objectiu de la càmera.

Una proposta de continuació del treball podria consistir en la reproducció del mateix ampliant-lo amb l'estudi de la profunditat (estudi de moviments compresos en les tres dimensions). Per tal que això sigui possible caldria capturar el moviment amb dues o més càmeres, o bé emprant càmeres d'infrarojos.

3. L'ús de marcadors en biomecànica

La biomecànica és una ciència que engloba aspectes d'anatomia, fisiologia, mecànica i cinesiologia. En la biomecànica s'apliquen els principis i lleis de la mecànica als éssers vius (en general als humans), per caracteritzar-ne els processos que s'hi desenvolupen. Aquests processos poden pertànyer a la mecànica de sòlids i músculs (musculoesquelètica) com la mecànica de líquids i gasos.

La finalitat de la biomecànica és la comprensió de disfuncions i patologies amb l'objectiu de deduir les actituds terapèutiques que se'n deriven. Així, algunes aplicacions de l'ús de la biomecànica són la pràctica esportiva, l'ergonomia i la fisioteràpia. Precisament aquesta última aplicació és la que s'explorava en la tesis de la Dra. Olga Borao d'on procedeix la idea per la realització del treball que ens ocupa.

En l'estudi de la cinemàtica de punts del cos humà s'utilitza una gran varietat de mètodes de mesura. D'entre les tècniques directes, que extreuen les dades directament del cos estudiat, el goniòmetre mesura les variacions en l'eix de rotació d'una articulació i els acceleròmetres l'acceleració del punt on s'apliquen.

Altres tècniques per la mesura de variables d'interès es basen en les imatges. Un gran nombre de les tècniques de mesura mitjançant imatges es centren en l'ús de marcadors, objectes que es col·loquen en els punts d'interès del cos que caracteritzen el moviment.

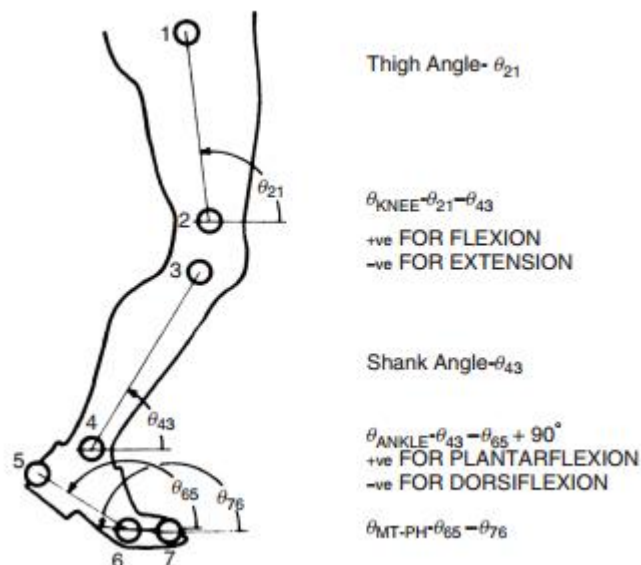


Fig. 3.1: Localització de marcadors i indicació dels angles que formen [2]

La cinematografia, per exemple, és un mètode del que ja s'escrivia el 1951 [4] i consisteix en la projecció dels fotogrames d'una pel·lícula en una pantalla dotada d'una graella. Allà es selecciona el marcador manualment i la posició d'aquest és transferida en un ordinador.

Aquesta tècnica ha sigut reemplaçada per la videografia, que constitueix una de les tècniques més usuals per tal de recollir dades cinemàtiques. En aquest sistema els marcadors utilitzats són reflectants, de manera que és senzill localitzar-los en la imatge.

Un altre sistema és l'optoelectrònica, on els marcadors estan connectats a un aparell electrònic per poder emetre polsos de manera seqüencial, els quals es detecten en una càmera especial.

Per altra banda, darrerament s'han desenvolupat mètodes que no requereixen de l'ús de marcadors. Aquests mètodes requereixen de càmeres que mesuren la profunditat de la imatge.

3.1. L'estudi de la cinemàtica en el treball

L'objectiu d'aquest treball, com s'ha elaborat en l'apartat corresponent, és l'obtenció de la posició de marcadors al llarg del desenvolupament d'un moviment, a partir del qual s'obté la variació de l'angle que formen dos o més marcadors. Com s'ha vist, generalment els marcadors s'empren per identificar alguna articulació a la que són solidaris. Així, es pot dir que el treball pretén donar informació sobre el moviment d'articulacions en la realització d'una activitat determinada. Per tant el que es desenvolupa és una eina de mesura cinemàtica per a l'ús en biomecànica que analitza el recorregut d'un punt del cos humà. D'aquest anàlisi, idealment, se n'extreu informació que s'usa com a suport en la valoració del problema que s'investiga.

En el cas de l'estudi de la Dra. Borao la variació d'angle i evolució de la posició dels marcadors es volia relacionar amb el grau d'activació de determinats grups musculars.

Si bé en els vídeos que s'han utilitzat com a referència per la realització del treball els marcadors emprats no són reflectants (Fig. 3.2), la metodologia que s'utilitza és semblant a la videografia.

En primer lloc es situen els marcadors als punts característics que es vol estudiar. A continuació es procedeix a la captura del moviment en vídeo. Un cop gravat el vídeo l'arxiu digital on s'emmagatzema es connecta a un ordinador, on mitjançant el software desenvolupat amb MATLAB s'obté el recorregut dels marcadors al llarg del vídeo i la variació d'angle entre aquests.





Fig. 3.2: Fotograma d'un dels vídeos emprats com a referència en el desenvolupament del treball on es visualitza els marcadors, identificant la pelvis. Addicionalment s'observa el muntatge de l'experiment. Font pròpia, del vídeo enregistrat per Olga Borao.

4. Processament d'imatges

En aquest projecte es treballa amb vídeos, sent necessari el tractament dels fotogrames que s'hi representen. Cadascun d'aquests fotogrames és una imatge digital, és a dir, la representació d'una imatge bidimensional per mitjà d'un nombre finit de punts anomenats píxels. Per la seva banda, cadascun dels píxels es representa amb un o més valors numèrics.

Per exemple, en el cas de les imatges en color (RGB) els píxels es representen per tres valors diferents que varien en funció de la quantitat de color vermell (R de red), verd (G de green) i blau (B de blue). Per altra banda, les imatges monocromàtiques consten de píxels que presenten un sol valor de gris que depèn de la intensitat.

El nombre de valors amb els que es caracteritza un píxel es correspon amb el nombre de canals de la imatge. Així, les imatges RGB són de 3 canals mentre que les d'escala de grisos en tenen un.

D'aquesta manera, cada canal de la imatge es pot representar mitjançant una matriu de dues dimensions composta per nombres reals (Fig. 4.1) de M files (alçada en píxels de la imatge) i N columnes (amplada).

$$f(p, q) = \begin{bmatrix} f(1, 1) & f(1, 2) & \dots & f(1, N) \\ f(2, 1) & f(2, 2) & \dots & f(2, N) \\ \vdots & \vdots & & \vdots \\ f(M, 1) & f(M, 2) & \dots & f(M, N) \end{bmatrix}$$

Fig. 4.1 : Funció bidimensional per accedir a píxels d'un canal de la imatge. Font pròpia.

Quan una imatge digital es modifica mitjançant una computadora es parla de processament d'imatges.

Una característica important de les imatges digitals és els bits utilitzats per caracteritzar cada valor del píxel (Fig. 4.2). El nombre de bits destinat a cada canal limita el nombre de valors que pot prendre el píxel en aquell canal. Així, l'ús d'un sol bit permet dos valors diferents (imatge binària), 8 bit en permet 256 i 16 bits fins a 65.536.



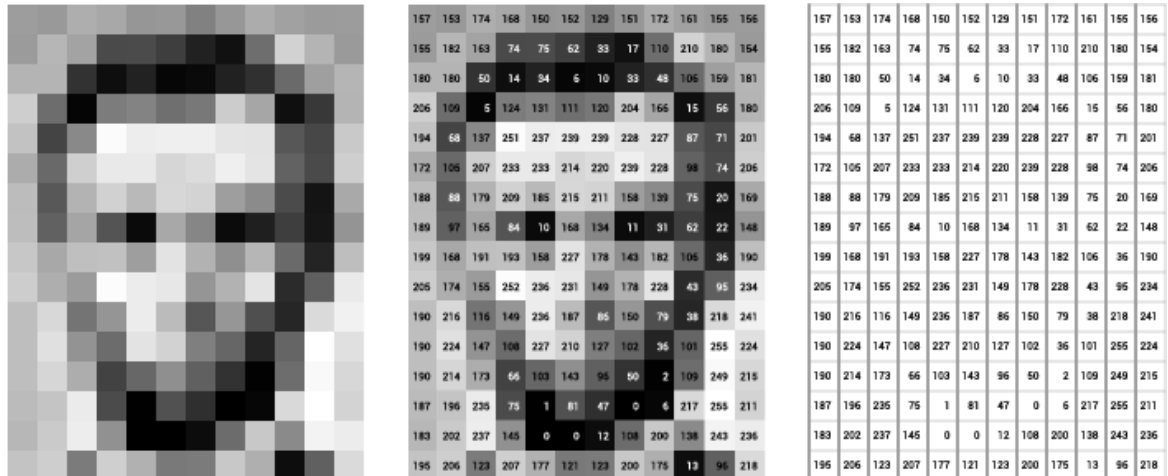


Fig. 4.2 Representació d'una imatge en escala de grisos, amb píxels de 8 bits [5].

En el treball es treballa amb canals de 8 bits, de manera que els valors d'un píxel per cada canal varien entre 0 i 255. En el cas de les imatges en escala de grisos, el 0 es correspon amb el color negre, mentre que el 255 correspon al blanc. En les imatges RGB, un valor de 255 en un canal significa que hi ha la màxima quantitat possible del color corresponent al canal. Al constar de tres canals de color diferents, a cada píxel d'una imatge RGB es destinen 24 bits (Fig. 4.3).



Fig. 4.3. Imatge en color (a) i els seus components R (b), G (c), i B (d) [6].

Els fotogrames dels vídeos utilitzats en el treball són del tipus RGB de 24 bits per píxel. Per tal de realitzar operacions matemàtiques necessàries per aconseguir l'objectiu del projecte és necessari treballar amb un sol canal, de manera que la imatge es passa a escala de grisos. Addicionalment es transforma la classe de dades dels píxels per treballar amb valors que varien entre 0,0 i 1,0 (classe de doble precisió) enlloc dels 0 i 255 que comporten els 8 bit per canal.



5. Metodologia pel seguiment dels marcadors

El seguiment d'un marcador al llarg d'un vídeo és la qüestió de més importància en el desenvolupament del programa, ja que sense ella no és possible l'assoliment de l'objectiu del treball. La definició del problema és senzilla, es tracta d'identificar un marcador en el fotograma inicial del vídeo i a partir d'aquest localitzar-lo en els següents fotogrames. Identificar el marcador consisteix en assignar-li la subimatge on queda representat (Fig. 5.1), obtenint així una matriu de píxels que es pot assumir de dimensions reduïdes (difícilment es generen més de 15 o 20 files o columnes).



Fig. 5.1: Identificació d'un marcador. Font pròpia.

La cerca de la localització d'una regió d'una imatge respecte el seu fotograma següent per tal d'obtenir el vector del moviment que s'hagi pogut succeir és un problema que històricament ha sigut investigat amb profunditat, i ho continua sent. Això és degut principalment al fet que en la codificació de vídeos l'estimació del moviment de diferent blocs en que es pot dividir una imatge és clau per la compressió dels vídeos. La compressió té lloc per mitjà de l'eliminació de redundàncies, i mitjançant l'estimació del moviment dels blocs és possible detectar redundàncies temporals. Per altra banda el seguiment de determinats patrons és necessari en multitud d'altres aplicacions.

En l'àmbit de l'estimació del moviment en imatges s'han desenvolupat diferents estudis que tracten el problema i que es poden classificar generalment en dues categories. Els mètodes per trobar el bloc que s'estudia es classifiquen en funció de si es basen en els píxels que conformen la imatge (mètodes directes) o bé en determinades característiques o punts d'interès d'aquesta (mètodes indirectes).

Els mètodes indirectes consisteixen en la detecció i extracció de trets característics en la regió de la imatge on es produeix el moviment a estudiar. Es comprova que aquestes característiques, com poden ser límits, cantonades, regions amb propietats diferenciades respecte la resta de la imatge, etc., es corresponen en cada fotograma, establint així si la seva posició ha variat. En el cas que ens ocupa, l'aplicació d'aquests mètodes ha resultat ineficaç. Es creu que els motius pels quals la detecció i extracció de característiques no ha resultat efectiva són per una banda la mida dels marcadors, donada pels vídeos emprats com a referència pel desenvolupament del treball, que resulta massa petita, i d'altra banda la qualitat de la imatge dels vídeos. Tot plegat impossibilita la detecció de característiques en el bloc de la imatge corresponent al marcador.

Per altra banda els mètodes directes prenen com a base un conjunt de píxels corresponents a una regió de la imatge anomenada bloc (Fig. 5.2). Per al desenvolupament del treball s'ha emprat un d'aquests mètodes, el *block matching* (coincidència de blocs), la utilització del qual és molt comuna en la codificació de vídeos, com a referent. Els algorismes que utilitzen el mètode *block matching* estimen la quantitat de moviment d'un bloc determinat o de cadascun dels blocs que s'estudien.



Fig. 5.2: Bloc d'una imatge amb la matriu de píxels corresponent. Font pròpia

Normalment el mètode empra algorismes d'estimació cap endarrere, és a dir, es defineixen els blocs a estudiar en un fotograma de referència i es busca la posició on es trobaven en el fotograma anterior [7]. Això és degut a que, com s'ha comentat anteriorment, l'aplicació principal del mètode és en la codificació de vídeos, en la que el fotograma de referència és dividit en la seva totalitat mitjançant els blocs. En el cas de cercar en el següent fotograma (estimació cap endavant) alguns blocs es podrien superposar a resultes del moviment (Fig. 5.3), amb la qual cosa la imatge a codificar no presenta zones que no estan definides per cap bloc.



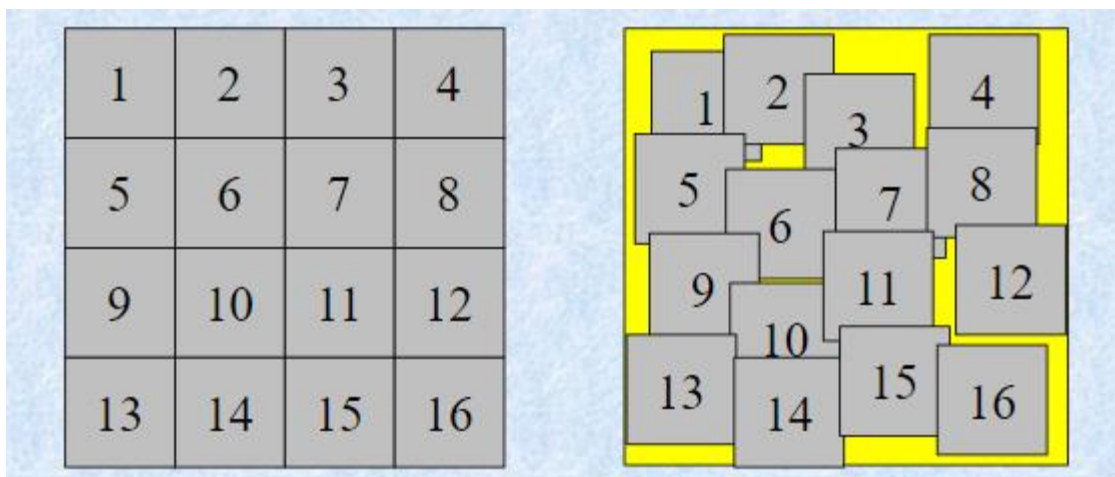


Fig. 5.3: Exemple del resultat d'una estimació cap endavant en una imatge dividida en 16 blocs **Error!**
No s'ha trobat l'origen de la referència..

En canvi, prenent com a base la imatge a codificar aquests problemes no sorgeixen.

De totes maneres, les dues possibilitats acaben trobant el mateix vector de moviment per a cada bloc. Així doncs, a efectes pràctics del treball qualsevol dels dos mètodes és equivalent. És per això que tot i que no és l'ús més habitual que se'n fa, en el desenvolupament d'aquest treball s'estudia el fotograma que ve a continuació del fotograma de referència enlloc del previ.

Així doncs, per cadascun dels blocs que es corresponen amb els marcadors i dels quals se'n vol determinar el moviment es troba un nou bloc en el fotograma següent, el qual coincideix amb el bloc original segons un criteri determinat.

5.1. Mètode de cerca

Com s'ha explicat anteriorment, un cop es coneix el bloc consistent en la subimatge del marcador del qual es vol fer el seguiment dins de la imatge de referència cal trobar-lo en el fotograma següent. Cal tenir en compte que tot i que el treball es centra en el seguiment de marcadors, la metodologia que en aquest apartat es descriu és aplicable a qualsevol tipus de patró sempre i quan aquest es trobi ben definit en el bloc original. És per això que a partir d'aquest moment es fa referència a marcadors i patrons indistintament.

La principal qüestió a considerar és el lloc on dur a terme la cerca del bloc en el nou fotograma. Una possibilitat és buscar el bloc en cadascun dels píxels del fotograma original.

Tenint en compte que la càmera amb la que es van gravar els vídeos emprats per l'elaboració d'aquest treball proporciona unes imatges de 384 píxels d'altura i 512

d'amplada, i considerant que el bloc del marcador és una matriu de 11x11 píxels, aquesta opció realitza la cerca en 502x374 píxels diferents. Això significa que s'ha d'implementar el criteri de comparació entre blocs 187.748 vegades. Queda clar que el cost computacional d'aquesta opció és molt elevat, i en el cas que el patró repetit en el fotograma podrien sorgir falses coincidències.

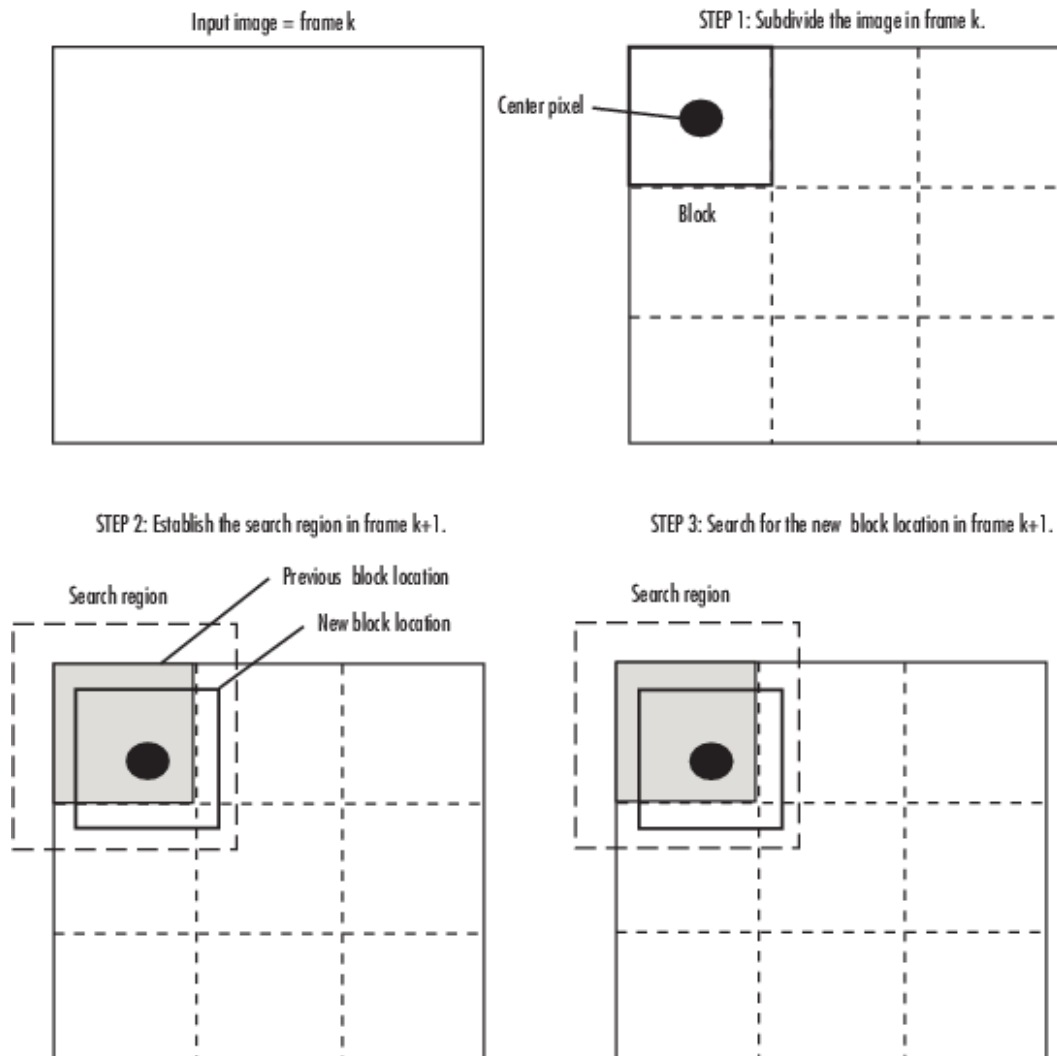


Fig. 5.4: Passos del mètode block matching. Font: MathWorks

De fet, en el temps que es porta investigant el mètode del *block matching* s'han desenvolupat multitud d'algoritmes per la recerca que tenen en compte la posició del bloc original (Fig. 5.4) en el fotograma de referència i que el seu principal objectiu és minimitzar el cost computacional.

5.1.1. Recerca exhaustiva

Un dels primers algoritmes en fer-se servir per la cerca de blocs és el de recerca exhaustiva



o completa. Consisteix en la cerca del bloc en el nou fotograma en cadascun dels píxels d'una regió d'interès delimitada. Al conèixer-se la posició inicial del bloc, és possible cercar-lo en una finestra en la que s'assumeix que es troba. A efectes pràctics, aquesta finestra es troba centrada en la posició inicial del patró, i la distància entre els límits del bloc i els de la finestra és el nombre de píxels que es suposa que el patró pot recórrer d'un fotograma al següent (Fig. 5.5). Cadascun dels blocs dins de la finestra es compara amb el bloc original segons un criteri determinat.

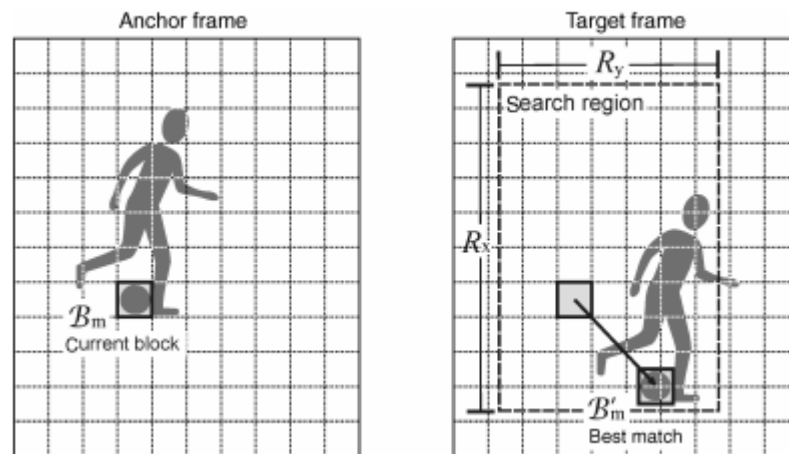


Fig. 5.5: Esquema del funcionament de la cerca exhaustiva [6].

Aquest mètode és el més senzill del conjunt d'algoritmes de cerca i també és el més fiable pel que fa al bloc trobat (sempre i quan es comparin amb el mateix criteri tots els possibles blocs). De totes maneres, el cost computacional continua sent elevat en comparació a algoritmes més elaborats.

5.1.2. Algoritmes sub-optimals

Amb l'objectiu de disminuir el cost computacional que comporta la realització de càlculs en un gran nombre de píxels es desenvolupen varis algoritmes de cerca que no contemplen la totalitat dels píxels en la regió d'interès, però no asseguren que el bloc trobat sigui el millor.

Generalment aquests mètodes consisteixen de diversos passos iteratius que acoten progressivament la posició del bloc en la finestra d'interès, i d'aquesta manera es compara un menor nombre de blocs [7].

L'algoritme de tres passos (Fig. 5.6), per exemple, compara en el primer pas nou blocs diferents centrats en píxels a una distància de quatre píxels entre si. Aquest primer pas pren com a origen dels càlculs la posició del bloc original. S'identifica el millor d'aquests blocs per mitjà del criteri de comparació determinat i s'hi centra la nova cerca, que es

realitzarà en píxels que l'envolten a una distància de dos píxels entre si. Finalment es realitza la mateixa operació en el tercer i definitiu pas, en el que els píxels comparats ja són contigus.

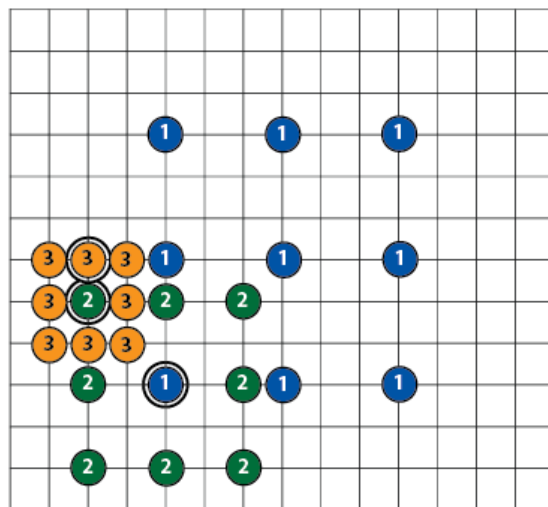


Fig. 5.6: Exemple de funcionament de l'algoritme de tres passos. Font: MathWorks

Altres algorismes que presenten un plantejament semblant són els següents:

- Cerca en quatre passos
- Cerca logarítmica bidimensional
- Cerca ortogonal
- Cerca en espiral
- Cerca de diamant

5.1.3. Algoritme de cerca emprat en el seguiment de marcadors

Per a la realització del treball s'ha emprat l'algoritme de la recerca exhaustiva, ja que és el que aporta el resultat de màxima qualitat. En el desenvolupament del programa s'ha considerat que la finestra on té lloc la cerca consta d'unes dimensions assumibles des del punt de vista del cost computacional, tenint en compte que el nombre de càlculs a realitzar creix de manera quadràtica a mesura que se n'incrementa l'amplada. Per altra banda, el cost computacional del programa es pot disminuir per altres mecanismes que es detallen més endavant.

5.2. Criteri de comparació

La comparació entre blocs es realitza mitjançant una funció de cost o de preu, que valora la similitud entre les dues imatges. Per cada bloc que s'estudia en la cerca es calcula el



resultat d'aplicar aquesta funció als dos blocs a comparar. Depenent del tipus de funció, el bloc òptim serà aquell que la minimitza o bé que la maximitza. Bàsicament les funcions de cost es divideixen en dos grups. Aquelles basades en la correlació i les que es basen en les diferències.

Els criteris basats en diferències mesuren l'error entre els píxels corresponents dels blocs, amb la qual cosa el bloc més semblant a l'original serà el que minimitza la funció. Els més populars són els següents:

$$C(x, y) = \sum_{i=1}^n \sum_{j=1}^n |F_a(i, j) - F_r(i + x, j + y)|$$

- Suma de diferències absolutes (error absolut)
- Suma de diferències al quadrat (error quadràtic)

$$C(x, y) = \sum_{i=1}^n \sum_{j=1}^n (F_a(i, j) - F_r(i + x, j + y))^2$$

- Mitjana de diferències absolutes (error absolut mitjà)

$$C(x, y) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n |F_a(i, j) - F_r(i + x, j + y)|$$

- Mitjana de diferències al quadrat (error quadràtic mitjà)

$$C(x, y) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (F_a(i, j) - F_r(i + x, j + y))^2$$

on n és l'amplada i alçada en píxels del bloc a comparar, F_a , i també del bloc de referència, F_r , x i y són les components del vector de desplaçament del bloc que s'està comparant, i C és el valor de la funció en el bloc.

Pel que fa als criteris basats en la correlació, cal destacar que la correlació creuada (producte escalar) entre dos vectors és una mesura de la seva similitud **Error! No s'ha trobat l'origen de la referència.**, per la qual cosa sembla lògic aplicar aquest sistema en la comparació dels blocs, i efectivament s'utilitza i és eficaç en un gran nombre de casos.

$$C_{fg} = \sum_{[i,j] \in R} f(i,j)g(i,j)$$

Tant els mètodes basats en diferències com la correlació creuada són subjectes a donar resultats equívocs (en el sentit que el bloc trobat no és el més semblant a l'original) quan la brillantor canvia al llarg de la gravació del vídeo, afectant així la intensitat dels píxels dels fotogrames.

La normalització dels píxels de les imatges f i g a comparar permet contrarestar aquest inconvenient (Fig. 5.7):

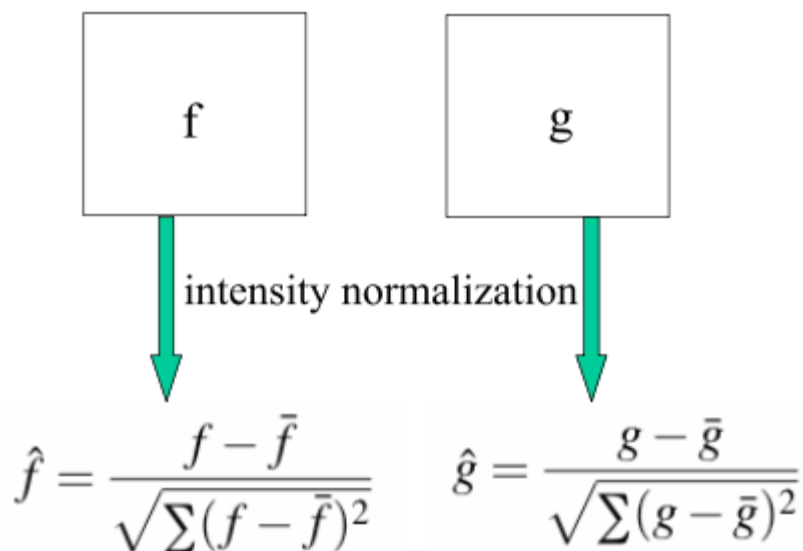


Fig. 5.7: Normalització de les imatges **Error! No s'ha trobat l'origen de la referència..**

Així, la correlació creuada normalitzada és un mètode molt robust i àmpliament aplicat en el *block matching*, de manera que s'ha decidit utilitzar-lo com a criteri de comparació en el programa desenvolupat.

5.3. Consideracions sobre la variació de l'objecte a seguir

En l'elaboració del mètode de seguiment pel cas concret del seguiment de marcadors, com s'ha exposat, s'ha emprat l'algoritme de la recerca exhaustiva per a trobar els diferents blocs candidats a correspondre's amb un bloc de referència, i aquests s'han comparat mitjançant el criteri de la correlació creuada normalitzada.

S'ha establert que el primer pas necessari per al seguiment del marcador és la identificació d'aquest en el primer fotograma del vídeo, obtenint el primer bloc a partir del qual es



localitzen els següents. Una qüestió que en els primers estadis del projecte es creia de gran importància en el seguiment del patró és la consideració del canvi de forma que aquest pot patir al llarg del moviment en el vídeo. Per un patró definit i robust del qual se sap que la seva forma no pateix cap variació es pot establir que no és necessària l'actualització del seu bloc al llarg del vídeo. D'aquesta manera s'estalvia el cost computacional de guardar la imatge corresponent al nou bloc cada vegada que es troba en el nou fotograma.

En l'inici del desenvolupament del programa aquesta tècnica no es va tenir en consideració, ja que els marcadors, al no tractar-se d'objectes simètrics o bidimensionals, canvien d'aspecte en el transcurs dels vídeos estudiats.

Es va pensar que en els casos en que es vol tenir en compte la variació del patró a mesura que aquest realitza un moviment, una solució raonable era l'actualització del bloc que el caracteritza cada vegada que es localitza en el fotograma corresponent segons el criteri de comparació establert.

En la realització del treball s'ha pogut comprovar que aquesta metodologia no sempre és adequada, podent implicar el fracàs de l'algoritme de seguiment.

Els motius usals d'una fallada en el seguiment del bloc es corresponen a la mala elecció dels paràmetres estudiats en els apartats anteriors d'aquest capítol. Per una banda un dimensionament insuficient de la finestra d'interès on es realitza la cerca pot provocar que el bloc es trobi fora d'aquesta. Per altra banda una mala elecció de l'algoritme de recerca pot implicar la falsa coincidència entre blocs. Finalment, un criteri que no consideri la diferència en la intensitat dels blocs a comparar pot comportar també mals resultats en el cas que la il·luminació de l'objecte a estudiar variï al llarg del vídeo.

Així doncs, s'ha pogut comprovar que l'actualització del bloc de referència també és s'ha de considerar com un paràmetre crític en el funcionament de l'algoritme de seguiment, però bàsicament pels problemes que pot ocasionar fer-la servir en el cas que el moviment del patró que s'estudia és lent.

Per moviments acusats, el canvi de posició dels píxels d'un fotograma respecte el següent són destacats. En canvi, en moviments que tenen lloc molt a poc a poc (els vídeos estudiats són gravats en càmera lenta) el moviment del patró d'un píxel al contigu pot produir-se al transcurs de varis fotogrames. D'aquesta manera, si es comparen dues imatges separades entre elles per n fotogrames és probable que s'identifiqui un canvi de posició del patró, mentre que si l'estudi es realitza comparant cadascun dels n fotogrames amb l'anterior les variacions seran massa lleus com perquè l'algoritme de seguiment consideri que hi ha hagut un canvi en la localització, encara que aquest s'hagi produït.

Un cop identificat el problema descrit s'ha procedit a realitzar la comparació dels blocs al llarg del vídeo respecte un bloc original fix i sense actualitzar-lo, tenint en compte que la posició que s'utilitza com a referència per a la recerca del marcador sí que es renova quan es localitza el bloc en el nou fotograma.

Per tal de no renunciar a la consideració de la variació de forma del marcador es proposen dues alternatives que no es desenvolupen en el programa però que a la pràctica haurien de proporcionar resultat equivalents al mètode aquí emprat.

Per una banda, el fet de no considerar tots els fotogrames del vídeo, sinó tractar-lo en intervals, considerant un fotograma i desestimant-ne un nombre a determinar dels que venen a continuació. El principal problema d'aquest mètode és la determinació del nombre de fotogrames que es desestimen, ja que cada vídeo representa un moviment amb una velocitat determinada. Per tant es corre el perill de que el bloc surti de la regió d'interès estudiada en l'algoritme de cerca en el cas que el nombre de fotogrames desestimats sigui massa gran. Mentre que en el cas de prendre un nombre reduït de fotogrames a descartar és possible que el problema continuï sent el mateix.

D'altra banda, l'actualització del bloc del marcador cada cop que la posició en el nou fotograma no es correspongui amb la del bloc trobat en el fotograma anterior. Així, cada cop que es detecta un moviment del marcador es pren com a bloc referència el nou bloc trobat. Aquest mètode és especialment útil ja que generalment el canvi de forma del marcador té lloc quan aquest es posa en moviment.

Finalment, cal destacar que en el cas que d'utilitzar una cerca sub-optimal, és a dir qualsevol algoritme que no desenvolupi una recerca exhaustiva, s'aconsella no actualitzar el bloc de referència ja que l'acumulació d'una seqüència de coincidències no òptimes pot implicar el mateix defecte



6. Desenvolupament del software i metodologia

6.1. Entorn de desenvolupament

Per la creació del software desenvolupat en aquest projecte s'ha treballat amb el programari MATLAB, en concret la versió 2016a, distribuïda gratuïtament per a estudiants de la UPC per part de MathWorks Inc. El MATLAB, acrònim de *matrix laboratory*, és a la vegada un entorn de computació numèrica i un llenguatge de programació.

6.2. Fases del desenvolupament

El primer disseny del programa ha tingut en compte les diferents funcions que aquest ha de realitzar per tal d'assolir el resultat que se n'espera: l'obtenció d'informació sobre el moviment dels marcadors. D'aquesta manera s'han establert un seguit de fases corresponents a les diferents funcions i que marquen l'estructura del programa ja que se succeeixen en sèrie. És a dir, és necessari que cadascuna de les fases s'hagi complert per tal de poder iniciar la següent. Així doncs les diferents fases no se solapen en el temps, però no són independents entre si degut a que cadascuna proporciona una informació que ha de ser utilitzada en la següent fase.

Pel disseny final s'ha tingut en compte la interacció de l'usuari amb el programa per tal que el primer pugui seleccionar la informació que desitja obtenir de manera senzilla i la rebí de manera clara. Les feines realitzades en aquesta fase es poden diferenciar entre aquelles que faciliten la interacció amb l'usuari i les que optimitzen el temps d'execució del programa.

6.3. Disseny inicial

Aquest primer disseny consisteix en el processament del vídeo per l'obtenció del recorregut d'un marcador determinat. Es tracta d'un codi estructurat i senzill que permet comprovar el correcte funcionament d'aquest primer programa, i suposa els fonaments del programa definitiu. El conjunt del codi presenta dos grans blocs diferenciats que s'esquematitzen en el següent esquelet (Fig. 6.1):

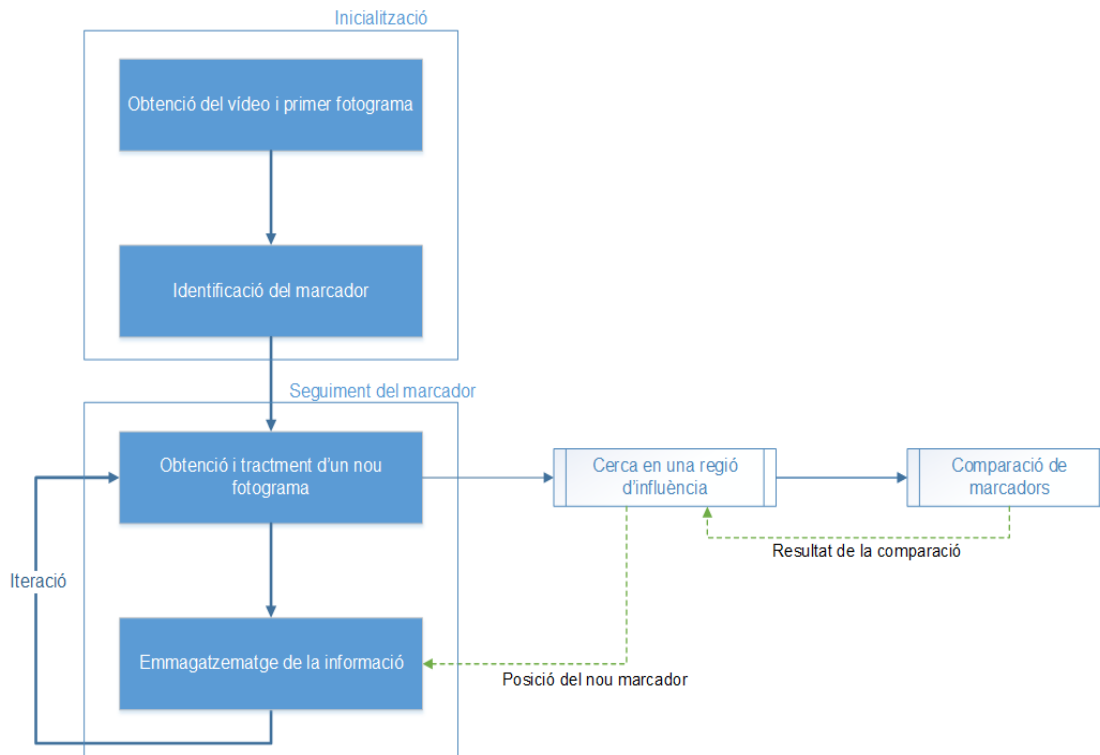


Fig. 6.1: Esquema del funcionament del programa. Font pròpia

6.3.1. Inicialització

El primer bloc en que s'estructura el programa és la inicialització de les variables més destacades necessàries al llarg de tot el programa. Es pot dividir en dues fases:

En primer lloc, s'obté el vídeo i s'extreu el primer fotograma habilitant-lo pel seu tractament posterior.

Obtenció del vídeo i del primer fotograma:

```

arxiu = 'TallCalaix2.mp4'; %per veure un altre vídeo s'ha de modificar
vidObj = VideoReader(arxiu);
base = readFrame(vidObj); %obtenció del primer fotograma
base = rgb2gray(base); %passem a un únic canal de grisos
base = im2double(base); %passem les dades a dobles per fer càlculs
  
```

Fig. 6.2: Codi del programa que s'encarrega de l'obtenció del vídeo i l'extracció del primer fotograma. Font pròpia.

Com es pot observar, en aquesta versió l'entrada del vídeo a estudiar es realitza dins del codi, amb la qual cosa per treballar amb un altre s'ha de modificar el programa. Per processar vídeos amb el MATLAB es crea un objecte mitjançant la funció *VideoReader*. Aquest objecte es pot tractar usant la funció *readFrame*, que proporciona el següent



fotograma disponible com a imatge. Com s'ha explicat en l'apartat 4, en funció del format de l'objecte la imatge resultant serà una matriu de dues o tres dimensions. Normalment els vídeos es graven en color, de manera que les imatges resultants són del tipus 'RGB', és a dir que tracten els colors en tres canals, i per tant la matriu és tridimensional amb una profunditat de tres unitats. És necessari doncs renunciar a aquest nivell d'informació i tractar les imatges com a objectes de dues dimensions mitjançant el pas de RGB a escala de grisos. Addicionalment, per tal de facilitar les operacions posteriors es transforma la informació corresponent als píxels de la matriu a doble precisió.

En la segona fase s'identifica el marcador i s'obté tant la seva posició relativa a la imatge inicial com la matriu que el caracteritza.

Identificació del marcador

```
imshow(base);  
hold on  
[x,y] = ginput(1); %posició del marcador  
amp_marc = 10; %distància entre els píxels dels extrems  
plotRectangle(x-amp_marc/2,y-amp_marc/2,amp_marc,amp_marc);  
drawnow;  
marcador = imcrop(base,[x-amp_marc/2,y-amp_marc/2,amp_marc,amp_marc]); %matriu del marcador
```

Fig. 6.3: Codi del programa destinat a la identificació del marcador. Font pròpia.

Es mostra la imatge per tal que l'usuari seleccioni la posició del marcador en la imatge. Aquesta posició es guarda per operacions posteriors. S'assigna una mida al marcador establerta per defecte i es mostra la seva situació dins del fotograma.

6.3.2. Seguiment del marcador

En el segon bloc és on té lloc el pes computacional del programa inicial, ja que consisteix en el seguiment del marcador al llarg de tot el vídeo. Es tracta d'un procés de cerca del marcador en el següent fotograma del vídeo que té lloc en una funció. Aquesta delimita una regió en la nova imatge centrada en la posició del marcador previ. Per cada punt d'aquesta regió es genera una subregió de les dimensions del marcador i es compara la subregió amb el marcador original per obtenir la posició de la que s'hi assembla més. La comparació es du a terme en una altra funció que realitza les operacions matemàtiques corresponents al mètode de comparació preferit. Aquest procediment té lloc de forma iterativa, obtenint la posició del marcador per cada fotograma del vídeo.

Seguiment del marcador

```
n=1;
while hasFrame(vidObj)
    base = readFrame(vidObj); %obtenció següent fotograma
    base = rgb2gray(base);
    base = im2double(base);
    [x(n+1),y(n+1)]=nouMarcador(x(n),y(n),marcador,base); %obtenció de la nova posició
    n=n+1;
    imshow(base)
    plotRectangle(x(n)-amp_marc/2,y(n)-amp_marc/2,amp_marc,amp_marc);
    drawnow; %comprovació que el seguiment és correcte
end
```

Fig. 6.4: Codi del programa que s'encarrega del seguiment del marcador. Font pròpia

En primer lloc s'inicia un comptador per identificar el fotograma del vídeo en el que es treballa. A continuació ja s'entra dins el procés iteratiu, que durarà sempre i quan l'objecte del vídeo tingui fotogrames disponibles. Aquesta condició ve donada per la funció *hasFrame*, que consulta l'existència de fotogrames disponibles per llegir. Així doncs, mentre no s'hagin tractat totes les imatges s'extreu la següent i es modifica per tal de poder-la estudiar i treballar-hi. La cerca de la nova posició del marcador té lloc en la funció *posicioPatro*. Aquesta accedeix a una altra funció, *comparaMatrius*, que compara el marcador original amb els candidats trobats en la cerca, i finalment retorna la posició de la imatge més semblant al marcador original, que s'emmagatzema juntament amb la resta de posicions obtingudes des del primer fotograma. A continuació s'actualitza el marcador i es mostra la posició del marcador relativa a la imatge per tal de comprovar visualment el correcte funcionament del codi.

6.4. Mètode de recerca

Com s'ha comentat, la recerca de la posició del nou marcador es duu a terme en la funció *posicioPatro*. Aquesta funció proporciona la nova posició d'un patró determinat donada la seva posició en el fotograma anterior, la imatge corresponent al patró i el nou fotograma. En el cas que ens ocupa, el patró que s'estudia és el marcador.

Tal i com s'ha explicat en l'apartat 5.1 de la memòria, l'algoritme de recerca de la posició del nou marcador consisteix en la recerca exhaustiva o completa.



```

% Cerca de la posició d'un patró determinat dins d'un fotograma
% a partir de la posició que tenia en el fotograma anterior
function [x1, y1] = posicioPatro(x0,y0,patro,fotograma)
[a,b]=size(patro); %altura i amplada (en pixels) del patró
max = -1; %el resultat de la comparació està acotat entre (-1,1)
lim = 5; %límit de pixels que assumim que es pot haver mogut en 1 frame
for i=-lim:lim
    for j=-lim:lim
        mat2 = imcrop(fotograma, [x0+i-(a-1)/2,y0+j-(b-1)/2,b-1,a-1]);
        %creació d'una matriu de les dimensions del patró
        res = comparaMatrius(patro,mat2);
        %valor de la NCC entre el patró inicial i el candidat a patró
        if res > max %per trobar la matriu qe mes sassembli al marcador
            max = res;
            x1 = x0+i; %posició del millor candidat
            y1 = y0+j;
        end
    end
end
end

```

Fig. 6.5: Codi de la funció que cerca el bloc en una finestra d'interès. Font pròpia.

En primer lloc s'obtenen les dimensions del patró, s'inicialitza la variable *max*, i s'estableix el límit de píxels que es considera que el patró es pot haver desplaçat tant vertical com horitzontalment. Així, s'estableix una regió on suposadament es troba el marcador, la finestra de interès. A continuació es recorre cadascun dels punts de la regió on es pot trobar el centre del marcador. En aquests punts es crea una nova matriu, *mat2*, de les dimensions del patró centrada en el punt en qüestió. Aquesta matriu correspon doncs a una subimatge del nou fotograma, i s'estudia com a candidat a ser correspondre's al nou marcador. Un cop obtingut el candidat a marcador cal comparar-lo amb el patró original per tal de saber quin és el seu grau de similitud. La comparació té lloc en la funció *comparaMatrius*.

6.5. Criteri de comparació

La funció *comparaMatrius*, com explicita el seu nom, s'encarrega de comparar dues matrius, corresponents a dues imatges, per tal de conèixer el seu nivell de semblança.

Existeixen diversos criteris per realitzar aquesta tasca, com s'ha comentat en l'apartat 5.2 d'aquesta memòria.

En el programa s'utilitza el mètode de la Correlació Creuada Normalitzada (NCC) com a

criteri de comparació entre dues imatges de mateixa mida. Com a resultat atorga un número entre -1 i 1. Si les dues imatges són idèntiques el resultat serà 1, i com menys similar siguin aquest valor anirà disminuint fins ser -1 en el cas que un imatge sigui l'oposada de l'altra.

```
% Comparació de dues matrius de mateix tamany amb NCC
function resultat = comparaMatrius(matrix1,matrix2)
m1 = mean2(matrix1);
s1 = std2(matrix1);
mat1_norm = (matrix1 - m1)./s1;
m2 = mean2(matrix2);
s2 = std2(matrix2);
mat2_norm = (matrix2 - m2)./s2;
resultat = mean(mat1_norm(:).*mat2_norm(:))...
    / (sqrt(mean(mat1_norm(:).^2)) * sqrt(mean(mat2_norm(:).^2)));
```

Fig. 6.6: Funció que realitza la correlació creuada normalitzada entre dos blocs. Font pròpia.

Com es pot veure, en aquesta funció el codi es limita a realitzar els càlculs pertinents a la correlació creuada normalitzada.

6.6. Disseny final

La versió definitiva del programa està dissenyada per tal que l'usuari en pugui fer ús sense haver-lo de modificar, i en compregui el funcionament sense haver de consultar el codi, tenir coneixements de programació o fins i tot estar familiaritzat amb l'entorn MATLAB. Es basa en l'estructura vista en el disseny inicial, i pel seguiment dels marcadors s'empren els mètodes que s'hi descriuen: la recerca exhaustiva per trobar els diferents candidats a marcador i la correlació creuada normalitzada per comparar els candidats amb l'original.

El disseny final enfoca el programa com una eina a ser utilitzada per l'usuari raó de ser, aquella persona que n'ha de fer ús i per a qui està destinat, que en aquest cas és l'investigador de biomecànica. En aquest sentit, es busca que el programa sigui fàcil de fer servir, és a dir que contingui unes instruccions entenedores, que sigui ràpid d'execució, i que informi a l'usuari del que pot haver fet malament en cas de fallada.

6.6.1. Inicialització

En la obtenció del vídeo es troba la primera millora del programa respecte la versió inicial, ja que es permet a l'usuari seleccionar qualsevol vídeo accessible des de l'ordinador des d'un cercador d'arxius, i en el cas que es no se'n seleccioni cap se surt del programa



donant un avís.

Obtenció del vídeo

```
[FileName,PathName] = uigetfile('*.avi');
assert(iscchar(FileName),'No has seleccionat cap vídeo')
arxiu = strcat(PathName,FileName);
input('Abans de començar, mira el vídeo i observa el fotograma en que comença el moviment.')
imshow(arxiu)
pause()
Inici_vid = input('\nFrame en que comença el moviment? ');
vidObj = VideoReader(arxiu);
```

Fig. 6.7: Codi del programa que sol·licita la interacció de l'usuari per a l'obtenció del vídeo a estudiar. Font pròpia.

Per altra banda, la interacció amb l'usuari també es fa palesa per mitjà d'instruccions que s'indiquen a la finestra de comandes. En primer lloc se li demana que identifiqui el fotograma on s'inicia el moviment per tal d'obviar els fotogrames inicials on no es desenvolupa cap mena d'activitat o fins i tot no s'hi representa el marcador, evitant un cost computacional innecessari.

6.6.2. Consideració de més d'un marcador

En el disseny final del programa s'ha habilitat la selecció de més d'un marcador per poder extreure informació de la seva posició relativa.

Realització de les operacions per cada marcador

```
num_Marc = input('\nNúmero de marcadors (enter) a seguir? ');
matx = [];
maty = [];
for i=1:num_Marc
```

Fig. 6.8: Codi del programa que permet l'anàlisi de més d'un marcador sense sortir del mateix. Font pròpia.

Així, es demana a l'usuari el nombre de marcadors que s'han d'estudiar, s'inicialitzen les variables on es guardaran les seves posicions, i es posa en marxa un procés iteratiu per estudiar cadascun dels marcadors per separat.

6.6.3. Extracció del primer fotograma i identificació del marcador

Pel cas de cada marcador, es selecciona com a primer fotograma a estudiar aquell

especificat per l'usuari anteriorment, i se'n extreu la imatge igual que en el disseny inicial. Per tal de minimitzar el cost computacional es demana a l'investigador si vol seleccionar una regió de la imatge de la qual el marcador no en surt en tot el vídeo.

Primer fotograma i identificació del marcador

```
vidObj.CurrentTime = Inici_vid/vidObj.FrameRate;
base = readFrame(vidObj);
base = rgb2gray(base); %passem a un únic canal de grisos
base = im2double(base); %passem les dades a doubles per fer calculs
area = questdlg('Vols seleccionar regió de interès? (recomenat)');
if strcmp(area,'Yes')
    figure('NumberTitle','off','Name','Selecciona la regio de interès');
    imshow(base)
    hold on
    [xs,ys]=ginput(2);
    rect=sort([xs,ys]);
    base = imcrop(base,[rect(1),rect(3),rect(2)-rect(1),rect(4)-rect(3)]);
    plotRectangle(rect(1),rect(3),rect(2)-rect(1),rect(4)-rect(3));
end
n=1;
figure('NumberTitle','off','Name','Selecciona el marcador')
imshow(base);
hold on
[x,y] = ginput(1);
amp_marc = 10; %es distancia entre pixels (amplada en pixels +1)
plotRectangle(x-amp_marc/2,y-amp_marc/2,amp_marc,amp_marc);
drawnow;
marcador = imcrop(base,[x-amp_marc/2,y-amp_marc/2,amp_marc,amp_marc]);
```

Fig. 6.9: Codi d'extracció del primer fotograma i identificació del marcador sobre aquest. Font pròpia

Finalment l'usuari és instat a identificar el centre del marcador, amb la qual cosa s'obté la seva posició inicial i el bloc original que serveix com a referència per la comparació de tots els blocs a estudiar posteriors.

6.6.4. Seguiment del marcador

Pel que fa al seguiment del marcador, el sistema és el mateix que es descriu en el disseny inicial. Una diferència rellevant en el funcionament de cara a l'usuari és la possibilitat d'escollir si es vol realitzar la visualització del seguiment del marcador. Si no es tria aquesta opció, el cost computacional serà considerablement menor ja que el programa no haurà de mostrar els centenars de fotogrames que conformen un vídeo.



Seguiment del marcador

```

visualitzacio = questdlg('Vols visualitzar el seguiment del marcador?');
if strcmp(visualitzacio,'Yes')
    close
    figure('NumberTitle','off','Name','Seguiment del marcador')
    imshow(base)
    hold on
end
while hasFrame(vidObj)
    base = readFrame(vidObj);
    if strcmp(area,'Yes')
        base = imcrop(base,[rect(1),rect(3),rect(2)-rect(1),rect(4)-rect(3)]);
    end
    base = rgb2gray(base);
    base = im2double(base);
    [x(n+1),y(n+1)]=nouMarcador(x(n),y(n),marcador,base);
    n=n+1;
    if strcmp(visualitzacio,'Yes')
        imshow(base)
        plotRectangle(x(n)-amp_marc/2,y(n)-amp_marc/2,amp_marc,amp_marc);
        drawnow;
    end
end

```

Fig. 6.10: Codi consistent del seguiment i actualització de la posició del marcador. Font pròpia

La funció *nouMarcador* és equivalent a la funció *cercaPosicio* vista en el disseny inicial. Donada la posició anterior, el bloc corresponent al marcador inicial i el fotograma o regió d'aquest que es vol estudiar, proporciona la nova posició en el nou fotograma del bloc més semblant a l'original. Empra també la cerca exhaustiva acotada a una finestra d'interès i compara els blocs pel criteri de la correlació creuada normalitzada. L'única diferència entre les funcions és l'estalvi pel que fa al temps de computació necessari, ja que es realitza la normalització del bloc del marcador original fora de la funció de comparació de blocs, de manera que no s'ha de realitzar per cada bloc a comparar. La nova funció de comparació és doncs *normCrossCorr*, que només ha de normalitzar el nou bloc a comparar i realitzar els càlculs de la correlació creuada.

6.6.5. Emmagatzematge dels resultats

Durant el seguiment del marcadors la posició canviant d'aquests es guarda en una variable que serà sobreescrita en el cas que el programa s'apliqui a més d'un marcador. És per això que és necessari l'emmagatzematge de les coordenades x i y en les variables inicialitzades anteriorment.

Emmagatzematge dels resultats

```
if strcmp(area, 'Yes')
    x=x+xs(1)-1;
    y=y+ys(1)-1;
end
close all
matx = [matx;x];
maty = [maty;y];
end
```

Fig. 6.11: Codi d'emmagatzematge de les posicions obtingudes del seguiment d'un marcador.
Font pròpia.

La variable *matx* consisteix d'una matriu que al finalitzar el seguiment de tots els marcadors presenta tantes files com marcadors s'hagin estudiat. En cada fila s'hi desa la posició del marcador en la coordenada x per cadascun dels fotogrames estudiats. Així, el nombre de columnes final serà igual al nombre de fotogrames que s'han analitzat. Per altra banda, la variable *maty* presenta les mateixes propietats, però emmagatzemant els valors de les coordenades y.

6.6.6. Tractament dels resultats

En aquest punt el programa ja ha obtingut les posicions dels marcadors al llarg del tram del vídeo que es vol estudiar. L'objectiu amb el qual s'ha desenvolupat el programa, però, no es limita a la situació de cada marcador sinó que pretén proporcionar més informació a l'investigador per tal d'analitzar el moviment que s'analitza.

Per aquest motiu es realitza un tractament de les posicions obtingudes, en concret el càlcul de la variació de l'angle entre els diferents marcadors, en cadascun dels fotogrames respecte l'angle que formaven en la posició inicial.



Càlcul de les variacions d'angle

```
totes_vars=[];
for i = 1:num_Marc-1
    for j = 1:num_Marc-i
        var = varAngle(matx(i,:),maty(i,:),matx(i+j,:),maty(i+j,:));
        var = var.*180/pi;
        totes_vars = [totes_vars;var];
        figure()
        xlabel('Fotograma')
        ylabel('Variació angular [graus]')
        plot(var)
    end
end
```

Fig. 6.12: Codi del programa encarregat de sol·licitar a una funció externa els valors de la variació d'angle entre cadascuna de les combinacions possibles de dos marcadors. Font pròpia.

El programa ha de considerar, en funció del nombre de marcadors que es volen estudiar, totes les combinacions possibles entre aquests per tal de calcular-ne la variació d'angle.

Si es considera cada marcador com un punt, sempre que se n'estudiï més d'un es pot traçar una línia entre dos dels marcadors. S'ha comprovat que a mesura que augmenta el nombre de marcadors, el vincles que es generen entre ells (el número de línies que els uneixen) augmenten en nombre de manera quadràtica seguint la sèrie triangular.

Així, quan es treballa amb dos marcadors s'estableix un vincle entre ells, quan es treballa amb tres se n'estableixen tres, amb quatre se'n formen sis, segons la taula:

Nombre de marcadors	Nombre de vincles
1	0
2	1
3	3
4	6
5	10
6	15
7	21

Taula 6.1: Nombre de variacions d'angle a estudiar en funció del nombre de marcadors. Font pròpia.

En concret, la sèrie es defineix per la fórmula:

$$T_n = \sum_{k=1}^n k = 1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$$

D'aquesta manera es determina el nombre total de relacions entre marcadors a estudiar. A la pràctica, per tal de facilitar la consideració dels vincles entre marcadors que ja s'han estudiat, s'estudia les relacions entre el primer marcador seleccionat i la resta, a continuació les del segon marcador amb els que s'han determinat després, i així successivament. Tenint en compte n marcadors i simbolitzant amb una x el vincle a estudiar, la següent figura clarifica el funcionament de la determinació dels vincles en el programa:

	Marcador 1	Marcador 2	Marcador 3	...	Marcador n-1
Marcador 2	x				
Marcador 3	x	x			
Marcador 4	x	x	x		
...	
Marcador n	x	x	x	x	x

Taula 6.2: Estructura del mètode seguit en el programa per a la determinació de les relacions entre marcadors a estudiar. Font pròpia.



Un cop determinades les relacions entre marcadors que s'han d'estudiar es procedeix a calcular la variació de l'angle al llarg del vídeo entre els marcadors.

```
function var_angle = varAngle(x1,y1,x2,y2)
    thetaini=asin((y2(1)-y1(1))/norm([x2(1)-x1(1),y2(1)-y1(1)]));
    for n=1:length(x1)
        thetanova=asin((y2(n)-y1(n))/norm([x2(n)-x1(n),y2(n)-y1(n)]));
        var_angle(n)=thetanova-thetaini;
    end
```

Fig. 6.13: Codi de la funció que realitza el càlcul de la variació angular al llarg del vídeo. Font pròpia.

El càlcul té lloc en la funció *varAngle*, que a partir del conjunt de coordenades x i y dels dos marcadors a estudiar proporciona la variació de l'angle entre ells en radians. Com s'observa en el codi, en primer lloc es cerca l'angle inicial que formen els marcadors. Per cada nova posició dels marcadors es calcula el nou angle i la diferència respecte l'angle inicial. Així s'obté la variació l'angle que formen els marcadors respecte la seva posició inicial en cada fotograma del vídeo.

Els resultats s'emmagatzemen, un cop convertits a graus, dins la variable *totes_vars* de la mateixa manera que es guardaven les posicions en les matrius de coordenades x i y. Finalment, per cada relació entre marcadors estudiada (per cada angle entre dos marcadors) es mostra el gràfic de l'evolució de la variació de l'angle al llarg dels fotogrames.

6.7. Validació del programa i resultats

La validació del programa es du a terme mitjançant l'execució d'aquest emprant els dos vídeos gravats durant els estudis de la Dra. Boreo. L'execució del programa s'ha dut a terme tenint en compte totes les combinacions possibles pel que fa a la selecció d'una regió d'interès on té lloc el moviment del marcador i la visualització del seguiment dels marcadors. La visualització del seguiment s'ha realitzat múltiples cops degut a la imprecisió que es genera en la selecció inicial del marcador, la qual cosa comporta que les posicions trobades en una validació respecte la següent puguin variar en un o dos píxels.

S'ha determinat que el programa funciona adequadament sempre i quan la selecció dels marcadors es realitzi de forma acurada.

Per als dos vídeos que s'han estudiat el programa representa l'evolució de la variació angular (Figs. 6.14 i 6.15), prenent com a marcador referència el primer que s'ha seleccionat:

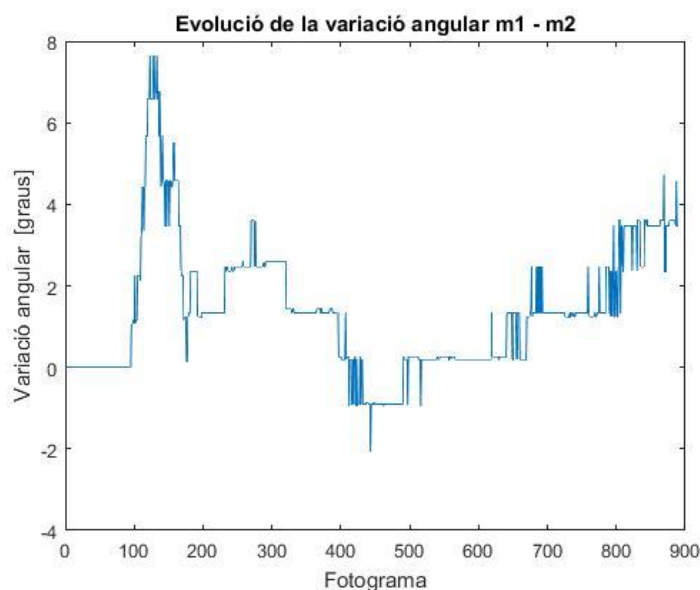


Fig. 6.14: Evolució de la variació de l'angle entre els marcadors en el primer vídeo. Font pròpia

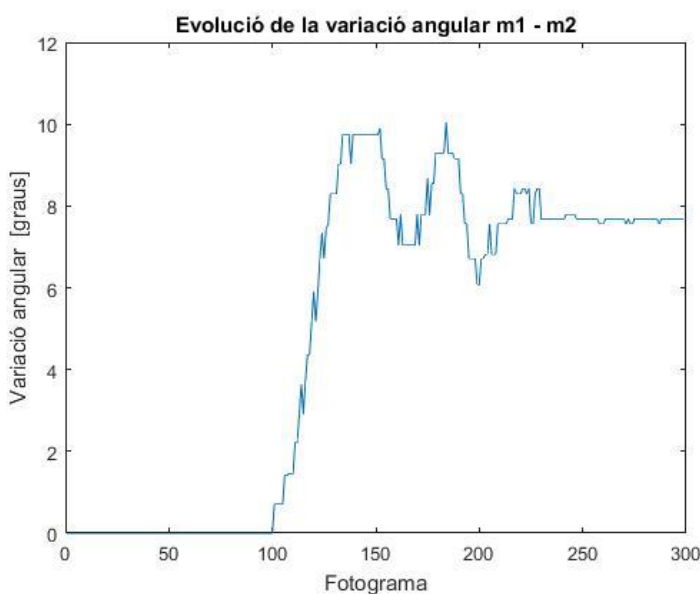


Fig. 6.15: Evolució de la variació de l'angle entre els marcadors en el segon vídeo. Font pròpia

Com es pot observar es mostra la variació angular en graus en cada fotograma estudiat.

No entra en els objectius del treball l'extracció d'informació sobre els gràfics, però pot facilitar la comprensió del que es visualitza la realització d'un breu comentari. Cal tenir en compte en comparar aquests dos gràfics que la duració de l'estudi del moviment en els dos casos és diferent, tot i que el moviment d'interès per la recerca presenta una duració semblant en els dos vídeos. Això es deu a que la gravació dels vídeos no acaba quan ho fa el moviment d'interès. És per aquest motiu que el segon vídeo presenta un major nombre



de dades. Es pot observar que en els dos vídeos estudiats la variació entre l'angle dels marcadors (i presumiblement també de les articulacions de la pelvis) s'inicia prop del fotograma número 100, la qual cosa no ha sigut provocada ni té rellevància. La variació angular augmenta de manera progressiva (casi de manera constant) fins arribar a un màxim local al cap d'uns 30 fotogrames de la primera variació. A partir d'aquest instant les articulacions dels dos pacients no actuen amb un mateix patró, amb la qual cosa es deixa per un expert en la matèria la valoració dels resultats.

6.8. Entorn de funcionament del programa

El software creat en aquest projecte consisteix d'un programa de Matlab executable des d'un ordinador personal. L'únic requeriment per tal de poder executar el programa sense que sorgeixin errors és que l'ordinador estigui equipat amb una versió del programari MATLAB, gratuït pels membres actius de la UPC (estudiants, PAS i PDI), superior a la 2014b.

L'espai de treball bàsic consta de quatre finestres (Fig. 6.16):

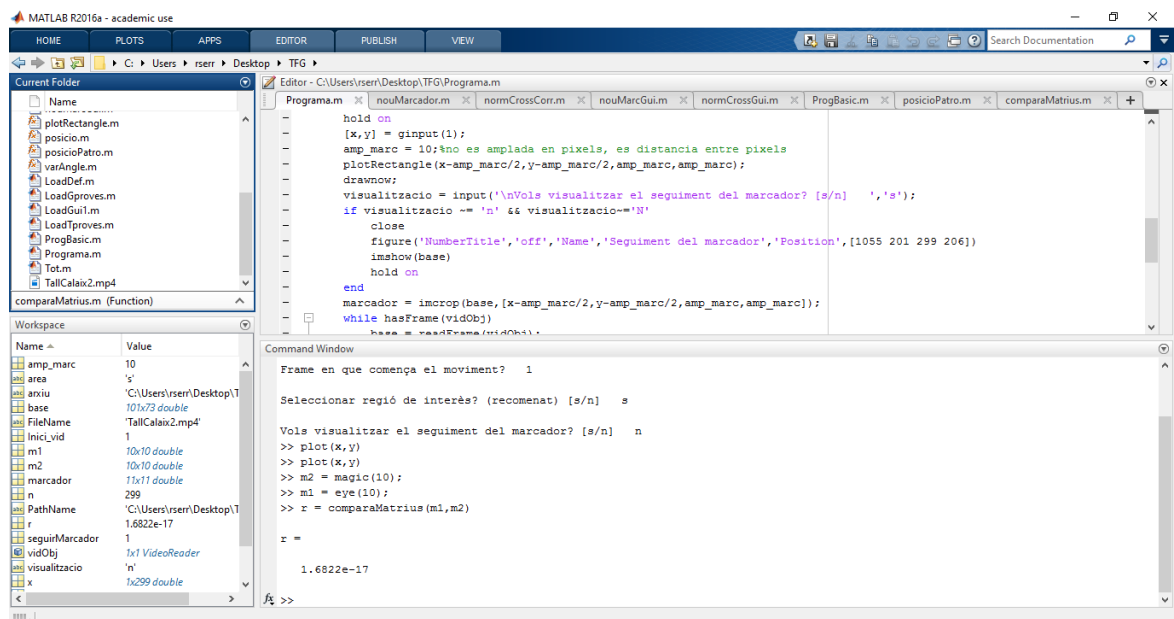


Fig. 6.16: Espai de treball del MATLAB. Font pròpia.

Per una banda la finestra *Current Folder* on es mostren els arxius presents a la carpeta on s'està treballant. Per tal d'executar el programa correctament, cal que els arxius *Executable.m*, *nouMarcador.m* i *normCrossCorr.m* es mostrin en aquesta finestra. Per tant, cal seleccionar la carpeta on es troben des del cercador de carpetes. En canvi, no és

necessari que el vídeo a estudiar es trobi en aquesta carpeta.

Els arxius presents a la finestra *Current Folder* són executables des de la finestra de comandes (*Command Window*) on es poden realitzar càlculs matemàtics, accedir a la informació d'objectes o funcions, modificar i visualitzar gràfics i imatges, etc. És des d'aquesta finestra que s'executa el programa, per la qual cosa només és necessari escriure'n el nom auto-explicatiu: «Executable». En aquesta mateixa finestra es guia a l'usuari pels diferents passos del programa.

A la finestra *Workspace* (espai de treball) es mostren les variables i objectes que s'han inicialitzat, i se'n mostra el seu valor o bé alguna referència a la seva mida o tipus.

Finalment la finestra *Editor* és on es realitzen programes i funcions, codi de varies files que s'ha d'utilitzar reiteradament. En una finestra d'aquestes característiques és on s'ha desenvolupat el programa.

Finalment s'ha de tenir en compte que els vídeos poden no ser reproduïbles segons el format en que estan gravats i el sistema operatiu en que es treballa. Així, en la següent taula (Fig. 6.17) s'observa una relació dels formats suportats pel programa en funció del sistema operatiu en el que s'executa:

VideoReader supports these file formats:

Platforms	File Formats
All Platforms	AVI, including uncompressed, indexed, grayscale, and Motion JPEG-encoded video (.avi) Motion JPEG 2000 (.mj2)
All Windows®	MPEG-1 (.mpg) Windows Media® Video (.wmv, .asf, .asx) Any format supported by Microsoft® DirectShow®
Windows 7 or later	MPEG-4, including H.264 encoded video (.mp4, .m4v) Apple QuickTime Movie (.mov) Any format supported by Microsoft Media Foundation
Macintosh	Most formats supported by QuickTime Player, including: MPEG-1 (.mpg) MPEG-4, including H.264 encoded video (.mp4, .m4v) Apple QuickTime Movie (.mov) 3GPP 3GPP2 AVCHD DV Note: For OS X Yosemite (Version 10.10) and later, MPEG-4/H.264 files written using <i>VideoWriter</i> , play correctly, but display an inexact frame rate.
Linux®	Any format supported by your installed plug-ins for GStreamer 0.10, as listed on http://gstreamer.freedesktop.org/documentation/plugins.html , including Ogg Theora (.ogg). Note: Support is provided for GStreamer 0.10 only, not later versions.

Fig. 6.17: Taula dels formats acceptats segons el sistema operatiu en que es treballa. Font: MatWorks

Si tot i aquesta informació no es té clar quin format de vídeo és compatible amb el programa, introduir la línia «VideoReader.getFileFormats» a la finestra de comandes del MATLAB proporciona un llistat dels formats acceptats per la funció VideoReader en el sistema operatiu utilitzat.



7. Pressupost

Els costos derivats d'aquest projecte es poden dividir entre aquells ocasionats pels recursos materials emprats en la realització del treball i els recursos humans que s'han destinat a la seva elaboració. Els recursos energètics no s'han tingut en consideració ja que el cost destinat a aquests és menyspreable.

A continuació es detalla el pressupost de les diferents partides que es deriven dels recursos utilitzats.

Pressupost dels recursos materials

En la Taula 7.1 es mostren les partides corresponents al material utilitzat per la realització del projecte. L'amortització es considera dins del període de realització del projecte, que ha sigut de 4 mesos.

Concepte	Preu unitari [€]	Unitats	Cost (€)
Ordinador portàtil HP	749	1	749
Ratolí	10	1	10
Lluminària	10	1	10
TOTAL			769,00 €

Taula 7.1: Pressupost dels recursos materials

Pressupost dels recursos humans

El nombre d'hores destinades al treball es correspon amb el nombre de crèdits ECTS del TFG, que és d'un total de 12 crèdits. Cada crèdit ECTS equival a un dedicació orientativa de 25 hores [11], amb la qual cosa s'obté un total de 300 hores. En la taula 7.2 es desglossa la destinació d'aquestes hores en tres blocs.

Per altra banda, s'ha considerat com a preu per hora el salari mínim d'un enginyer tècnic **Error! No s'ha trobat l'origen de la referència..**

Concepte	Preu unitari [€/h]	Unitats [h]	Cost [€]
Estudi previ i disseny	10,34	50	517
Desenvolupament del software	10,34	200	2.068
Redacció	10,34	50	517
TOTAL			3.102,00 €

Taula 7.2: Pressupost dels recursos humans

Pressupost global

Finalment les diferents partides s'ajunten en el pressupost global, que considera totes les despeses del projecte (Taula 7.3).

Concepte	Cost [€]
Recursos materials	769
Recursos humans	3102
TOTAL	3.871,00 €

Taula 7.3: Pressupost total



8. Impacte ambiental

En aquest apartat s'identifiquen els efectes sobre el medi ambient que es deriven del projecte. Aquests efectes es poden dividir entre aquells que han sigut causats durant el desenvolupament del treball i els que sorgeixen durant l'ús de l'eina creada.

Les activitats desenvolupades en aquest projecte cauen bàsicament dins de l'àmbit de la informàtica, concretament en el desenvolupament de software. Això fa que l'impacte que té sobre el medi ambient sigui de petites dimensions, ja que no es genera cap tipus de contaminant de manera directa. De totes maneres, tant en el procés inicial de documentació com en el desenvolupament del software i l'elaboració de la memòria s'ha consumit electricitat, amb la qual cosa s'han generat emissions de CO₂ de manera indirecta. Aquestes emissions es detallen a continuació:

Element de consum	Temps de consum [h]	Potència consumida [W]	Emissió de CO₂ [kg]
Ordinador	300	250	22,65
Elements d'il·luminació	230	100	6,95
TOTAL EMISSIONS DE CO₂			29,60

Taula 8.1: Emissions de CO₂ generades durant el projecte

Pel càlcul de les emissions de CO₂ es considera el mix de producció bruta d'energia elèctrica general recomanat per l'OCCC (Oficina Catalana pel Canvi Climàtic), que per l'any 2015 va ser de 302 g CO₂/ kWh [13].

Pel que fa a l'ús de l'eina que s'ha desenvolupat en el treball, aquest tampoc genera residus de manera directa, però també és necessari el consum d'energia, amb les emissions de CO₂ que se'n deriven. En aquest cas no es coneix el nombre d'hores que es destinaran a l'ús del programa, de manera que no es pot calcular l'emissió del diòxid de carboni a l'atmosfera.

Per altra banda, els material emprats en l'elaboració del treball i en l'ús de l'eina, com són els ordinadors i les lluminàries usades, tenen una vida útil llarga, amb la qual cosa si se'n fa un reciclatge adequat l'impacte sobre el medi ambient serà molt reduït.

Finalment es té en compte l'impacte sobre l'entorn social i econòmic. Considerant que la finalitat del projecte és de caràcter informatiu i pel seu ús en la investigació, es determina que l'impacte que genera en aquest entorn és positiu.



Conclusions

A partir de la feina realitzada en aquest treball de fi de grau s'ha aconseguit desenvolupar un programa de seguiment de marcadors al llarg d'un vídeo per a l'ús en biomecànica, executable en un ordinador personal.

S'ha verificat que el mètode de la coincidència de blocs aporta un bon resultat al seguiment de marcadors, concretament quan s'utilitza en combinació amb un algoritme de cerca exhaustiu i la comparació pel criteri de la correlació creuada normalitzada.

S'ha comprovat la necessitat de realitzar una bona identificació dels marcadors per assegurar-ne el correcte seguiment i la minimització d'imprecisions en els càlculs.

Finalment, s'ha vist que l'actualització del bloc de referència del marcador no sempre suposa una millora del mètode de seguiment encara que la forma del marcador no es mantingui constant al llarg d'un vídeo.

Així doncs, s'ha comprovat que no és necessària la utilització de marcadors reflectants ni càmeres i software especialitzat per tal de realitzar un seguiment del marcador fiable i extreure dades significatives d'un moviment.

Per a futures línies de treball es considera que es podria estudiar la posició de marcadors en tres dimensions. Un altre estudi d'interès pot ser la realització d'una comparativa entre els resultats obtinguts mitjançant diferents algoritmes de cerca de la posició del marcador i diferents criteris de comparació.

Agraïments

Al professor Antoni Susín, director del treball de grau, per haver ofert aquesta proposta de treball i haver estat en disposició en tot moment.

Bibliografia

Referències bibliogràfiques

- [1] BORAO, OLGA. *Musculatura estabilitzadora de l'extremitat inferior: estudi de la influència de les sinergies musculars de la cama, en relació a la patologia del turmell*. Lleida, 2015.
- [2] C. Eechaute, P. Vaes, W. Duquet, B. Van Gheluwe. *Test-Retest Reliability of Sudden Ankle Inversion Measurements in Subjects With Healthy Ankle Joints*. Journal of Athletic Training: 2007;42(1):60–65.
- [3] WINTER, D. A.: *Biomechanics and Motor Control of Human Movement*. Ontario: 2009.
- [4] EBERHART, H.D., INMAN V.T. *An evolution of experimental procedures used in a fundamental study of human locomotion*. Annals of the New York Academy of Sciences: 1951.
- [5] LEVIN, G. *Image Processing and Computer Vision*. [http://openframeworks.cc/ofBook/chapters/image_processing_computer_vision.html]
- [6] MARQUES, O. *Practical Image and Video Processing Using MATLAB*. Florida: 2011.
- [7] TURAGA, D., ALKANHAL, M. *Search Algorithms for Block-Matching in Motion Estimation*. Pittsburg: 1998. [http://www.ece.cmu.edu/~ee899/project/deepak_mid.htm, 17 d'octubre de 2016].
- [8] HU, Y.H. *Introduction to Image and Video Coding Algorithms*. [<http://slideplayer.com/slide/9415253>, 17 d'octubre de 2016]
- [9] PESQUET-POPESCU, B., CAGNAZZO, M., DUFAUX, F., *Motion Estimation Techniques*. París. [https://cagnazzo.wp.mines-telecom.fr/files/2013/05/poly_me.pdf, 17 d'octubre de 2016].
- [10] COLLINS, R. *Correspondence Matching*. Pennsilvània [<http://www.cse.psu.edu/~rtc12/CSE486/lecture07.pdf>, 17 d'octubre de 2016]
- [11] UNIVERSITAT POLITÈCNICA DE CATALUNYA, ETSEIB. *Normativa de Treballs de Fi de Grau*. Barcelona 2016. [http://www.etseib.upc.edu/docs/pfc/TFG_Normativa-junta-11-07-16_aprovat-1.pdf, 17 d'octubre de 2016].

- [12] MINISTERIO DE EMPLEO Y SEGURIDAD SOCIAL. *Resolución de 12 de febrero de 2013, de la Dirección General de Empleo, por la que se registran y publican las tablas salariales para 2013 del I Convenio colectivo de Dematic Logistic Systems, SA*. BOE 27 de febrer de 2013.
- [13] OFICINA CATALANA DEL CANVI CLIMÀTIC. *Nota Informativa sobre la Metodologia d'Estimació del Mix Elèctric per part de l'OCCC*. Febrer 2016. [http://canviclimatic.gencat.cat/web/.content/home/reduex_emissions/factors_emissio_associats_energia/160219_Nota-metodologica-mix_cat.pdf, 17 d'octubre de 2016].
- [14] O. Borao, A. Planas, A. Susín, F. Corbi. *Ankle muscle electromyographic activity under the influence of a low intensity isometric contraction of the lateral hip muscles*. Submitted to Journal of Electromyography and Kinesiology. 2015

Bibliografia complementària

DUFOUR, M., PILLU, M. *Biomecánica funcional*. París: 2006